



**TUGAS AKHIR - TE 141599**

**PERANCANGAN KONTROLER PI *SELF-TUNING* UNTUK  
SISTEM PENGATURAN TEMPERATUR PADA TANGKI  
TIMBUN CPO**

Puji Rahayu  
NRP. 2214 105 072

Dosen Pembimbing  
Dr. Ir. Mochammad Rameli  
Imam Arifin, ST., MT.

JURUSAN TEKNIK ELEKTRO  
Fakultas Teknologi Industri  
Institut Teknologi Sepuluh Nopember  
Surabaya 2016



***FINAL PROJECT - TE 141599***

***PI SELF-TUNING CONTROLLER DESIGN FOR  
TEMPERATURE CONTROL SYSTEM ON CPO STORAGE  
TANK***

Puji Rahayu  
NRP. 2214 105 072

*Supervisor*  
Dr. Ir. Mochammad Rameli  
Imam Arifin, ST., MT.

***DEPARTEMENT OF ELECTRICAL ENGINEERING  
Faculty of Industrial Technology  
Institut Teknologi Sepuluh Nopember  
Surabaya 2016***

**PERANCANGAN KONTROLER PI *SELF-TUNING* UNTUK  
SISTEM PENGATURAN TEMPERATUR PADA TANGKI  
TIMBUN CPO**

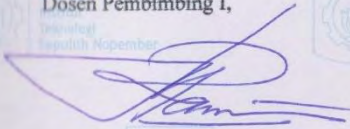
**TUGAS AKHIR**

Diajukan Guna Memenuhi Sebagian Persyaratan Untuk  
Untuk Memperoleh Gelar Sarjana Teknik  
Pada

Bidang Studi Teknik Sistem Pengaturan  
Jurusan Teknik Elektro  
Institut Teknologi Sepuluh Nopember

Menyetujui

Dosen Pembimbing I,



Dr. Ir. Mochammad Rameli  
NIP. 195412271981031002

Dosen Pembimbing II,



Imam Arifin ST., MT.  
NIP. 197302222002121001



# PERANCANGAN KONTROLER PI *SELF-TUNING* UNTUK SISTEM PENGATURAN TEMPERATUR PADA TANGKI TIMBUN CPO

Puji Rahayu  
2214 105 072

Dosen Pembimbing I : Dr. Ir. Mochammad Rameli  
Dosen Pembimbing II : Imam Arifin ST., MT.

## ABSTRAK

Sistem pengaturan proses di industri biasanya menerapkan kontroler *Proportional Integral Derivative* (PID) karena ketahanannya terhadap gangguan. Pada saat sistem sedang beroperasi, parameter *plant* dapat berubah tanpa bisa diprediksi. Perubahan tersebut menyebabkan parameter kontroler PID tidak lagi sesuai dengan kondisi *plant* dan apabila diabaikan, performa sistem tidak akan memenuhi kriteria yang diinginkan. Oleh karena itu, dibutuhkan proses *update* parameter kontroler PID agar sesuai dengan perubahan yang terjadi pada *plant*. Pada tangki timbun *Crude Palm Oil* (CPO) dibutuhkan temperatur 50°C sampai 55°C untuk menjaga kualitas CPO yang tersimpan dan agar pembacaan sensor aliran saat proses *unloading* lebih akurat. Penerapan metode PI *self-tuning* pada sistem ini bertujuan untuk mengatur temperatur tangki tersebut supaya dapat selalu memenuhi jangkauan yang diinginkan. Sistem dengan kontroler yang diusulkan tersebut mampu melakukan adaptasi saat terjadi perubahan parameter *plant* dan dapat mencapai keadaan tunak lebih cepat sesuai *set point* yang diberikan daripada sistem dengan kontroler PI konvensional.

**Kata Kunci :** Tangki timbun, CPO, PI *self-tuning*

# SELF-TUNING PID CONTROLLER DESIGN FOR TEMPERATUR CONTROL SYSTEM ON CPO STORAGE TANK

Puji Rahayu  
2214 105 072

Dosen Pembimbing I : Dr. Ir. Mochammad Rameli  
Dosen Pembimbing II : Imam Arifin ST., MT.

## ***ABSTRACT***

*Industrial process control system usually applied Proportional Integral Derivative (PID) controller because of its robustness with respect to disturbance. While the system was operating, the parameter of the plant could change without any prediction. The changes caused the PID parameters couldn't match with the plant's condition and if we didn't respected to it, the system's performance couldn't follow the desired criteria. Therefore, the system needs to update the process parameters for PID controller, so it could be matched with the change of plant's condition. Tank farm Crude Palm Oil (CPO) needs temperature range between 50°C-55°C to keep the quality of saved CPO and in order to the reading of flow sensor while unloading process be more accurate. Application of PI self-tuning method of this system to control the tank temperature, so that it could be as the input. The system with the proposed method could do the adaptation when the change of the plant's parameters has occurred and it could achieved steady state which appropriate to the set point that has given, faster than the system with conventional PI controller.*

**Keyword :** Tank farm, CPO, PI self-tuning.

## DAFTAR ISI

HALAMAN JUDUL .....	ii
PERNYATAAN KEASLIAN TUGAS AKHIR.....	iii
LEMBAR PENGESAHAN .....	v
ABSTRAK.....	vii
<i>ABSTRACT</i> .....	ix
KATA PENGANTAR .....	xi
DAFTAR ISI .....	xiii
DAFTAR GAMBAR.....	xv
DAFTAR TABEL .....	xvii
BAB 1 PENDAHULUAN .....	1
1.1 Latar Belakang.....	1
1.2 Perumusan Masalah .....	2
1.3 Batasan Masalah .....	2
1.4 Tujuan Penelitian .....	3
1.5 Metodologi .....	3
1.6 Sistematika Penulisan .....	3
1.7 Relevansi .....	4
BAB 2 SISTEM PENGATURAN TEMPERATUR DAN METODE ADAPTASI .....	5
2.1 Sistem Pengaturan Temperatur .....	5
2.1.1 Sistem Tangki .....	5
2.1.2 Sensor <i>Resistance Temperature Detector</i> (RTD) .....	6
2.2 Kontroler PID .....	7
2.3 Identifikasi <i>Plant</i> .....	10
2.3.1 <i>Pseudo Random Binary Sequence</i> (PRBS) .....	10
2.3.2 Metode Identifikasi <i>Least Squares</i> .....	11
2.4 Metode Validasi.....	13
2.5 Metode <i>Self-Tuning Regulator</i> (STR) .....	14
2.5.1 Estimasi Parameter .....	15
2.5.2 Perancangan Kontroler .....	15
2.6 Simulasi <i>Hardware in the Loop</i> .....	16
BAB 3 PERANCANGAN SISTEM.....	19
3.1 Sistem Tangki Timbun CPO.....	19
3.2 Pemodelan <i>Plant</i> .....	19
3.2.1 Pemodelan <i>Plant</i> Tangki Timbun .....	20

3.2.2	Pemodelan Sensor RTD .....	23
3.2.3	Fungsi Alih Keseluruhan .....	24
3.3	Identifikasi <i>Plant</i> .....	24
3.4	Validasi Model.....	27
3.5	Perancangan Kontroler.....	28
3.5.1	Perancangan Kontroler PI .....	29
3.5.2	Perancangan Metode Estimasi Parameter .....	31
3.5.3	Perancangan Metode <i>Self-Tuning Regulator</i> .....	31
3.6	Perancangan Simulasi <i>Hardware in the Loop</i> .....	32
BAB 4	PENGUJIAN DAN ANALISA .....	35
4.1	Pengujian Program untuk Membangkitkan Sinyal PRBS .....	35
4.2	Pengujian Sistem dengan Kontroler PI .....	36
4.2.1	Pengujian Sistem dengan Kontroler Proporsional.....	36
4.2.2	Pengujian Sistem dengan Kontroler Proporsional Integrator.....	37
4.2.3	Pengujian Sistem dengan Kontroler PI untuk Variasi $K_p$ .....	39
4.2.4	Pengujian Sistem dengan Kontroler PI untuk Variasi $\tau_i$ .....	40
4.2.5	Pengujian Sistem dengan Kontroler PI Ditambahkan Gangguan .....	40
4.2.6	Pengujian Sistem dengan Kontroler PI Ditambahkan <i>Noise</i> .....	42
4.3	Pengujian Sistem dengan Kontroler PI <i>Self-Tuning</i> .....	43
4.4	Simulasi <i>Hardware in the Loop</i> .....	44
BAB 5	PENUTUP .....	47
	DAFTAR PUSTAKA .....	49
	LAMPIRAN.....	51

## DAFTAR GAMBAR

Gambar 2.1	<i>Gas Tank</i> .....	6
Gambar 2.2	<i>Fuel Tank</i> .....	6
Gambar 2.3	Sensor RTD TMT142R.....	7
Gambar 2.4	Diagram blok kontroler PID.....	7
Gambar 2.5	Alur untuk membangkitkan sinyal PRBS.....	10
Gambar 2.6	Sinyal PRBS.....	11
Gambar 2.7	Diagram blok <i>self-tuning regulator</i> .....	14
Gambar 2.8	Prinsip simulasi <i>Hardware in the Loop</i> .....	17
Gambar 3.1	Diagram pemodelan sistem pengaturan temperatur .....	20
Gambar 3.2	P&ID sistem pengaturan temperatur tangki timbun .....	20
Gambar 3.3	Diagram blok sistem.....	24
Gambar 3.4	Diagram blok proses identifikasi.....	24
Gambar 3.5	Hasil identifikasi dengan masukan <i>unit step</i> .....	25
Gambar 3.6	Hasil identifikasi dengan masukan PRBS .....	26
Gambar 3.7	Respon sistem <i>open loop</i> .....	29
Gambar 3.8	Diagram blok sistem dengan kontroler PI .....	30
Gambar 3.9	Konfigurasi komputer <i>virtual plant</i> dan kontroler .....	33
Gambar 3.10	Konfigurasi perancangan simulasi sistem .....	33
Gambar 4.1	Sinyal PRBS.....	35
Gambar 4.2	Diagram blok sistem dengan kontroler proporsional.....	36
Gambar 4.3	Respon sistem dengan kontroler proporsional.....	37
Gambar 4.4	Diagram blok sistem dengan kontroler PI .....	38
Gambar 4.5	Respon sistem dengan kontroler PI .....	38
Gambar 4.6	Respon sistem dengan kontroler PI untuk variasi $K_p$ .....	39
Gambar 4.7	Respon sistem dengan kontroler PI untuk variasi $\tau_i$ .....	40
Gambar 4.8	Diagram blok sistem dengan kontroler PI ditambahkan gangguan .....	41
Gambar 4.9	Respon sistem dengan kontroler PI ditambahkan gangguan .....	41
Gambar 4.10	Diagram blok sistem dengan kontroler PI ditambahkan <i>noise</i> .....	42
Gambar 4.11	Respon sistem dengan kontroler PI ditambahkan <i>noise</i> ...	43
Gambar 4.12	Respon sistem dengan kontroler PI <i>Self-Tuning</i> .....	44
Gambar 4.13	Respon sistem di komputer kontroler .....	45
Gambar 4.14	Respon sistem di komputer <i>virtual plant</i> .....	45



## DAFTAR TABEL

Tabel 3.1	Parameter pemodelan <i>plant</i> .....	21
Tabel 3.2	Parameter <i>plant</i> tangki timbun .....	23
Tabel 3.3	Hasil identifikasi dengan masukan <i>unit step</i> .....	26
Tabel 3.4	Hasil identifikasi dengan masukan PRBS .....	27
Tabel 3.5	Validasi model dengan masukan identifikasi <i>unit step</i> .....	27
Tabel 3.6	Validasi model dengan masukan identifikasi PRBS .....	28

## **BAB 5**

### **PENUTUP**

Berdasarkan penelitian yang telah dilakukan dapat disimpulkan bahwa sistem pengaturan proses di industri memiliki parameter-parameter yang dapat berubah sewaktu-waktu tanpa dapat diprediksi. Kondisi tersebut dapat diatasi dengan metode pengaturan *self-tuning*, dimana parameter kontroler diperbarui terus-menerus saat terjadi perubahan pada *plant*. Kontroler PI *self-tuning* yang diterapkan pada penelitian ini dapat bekerja sesuai tujuan perancangannya yaitu pada saat terjadi perubahan parameter *plant*, sistem dapat melakukan adaptasi sampai kontroler memperoleh parameter yang sesuai dengan kondisi *plant* yang telah mengalami perubahan. Proses adaptasi disini ditunjukkan dengan respon sistem yang mengalami osilasi hingga kemudian kembali ke keadaan tunak saat parameter kontroler baru yang sesuai telah diperoleh.

Saat nilai *set point* sistem dirubah, seharusnya respon sistem dengan kontroler PI *self-tuning* tidak lagi mengalami osilasi karena tidak ada perubahan parameter *plant*, artinya tidak ada proses adaptasi yang terjadi. Namun kasus pada penelitian ini, respon sistem masih mengalami sedikit *overshoot* seperti halnya sistem model orde 2. Hal tersebut masih dapat ditoleransi, karena *overshoot* yang terjadi tidak begitu besar. Diharapkan untuk penelitian selanjutnya, *overshoot* tersebut dapat diminimalisir atau bahkan dihilangkan dengan menerapkan metode pengaturan tertentu.

## **BAB 3**

### **PERANCANGAN SISTEM**

#### **1.1 Sistem Tangki Timbun CPO**

Tangki timbun merupakan *plant* yang biasa dibangun untuk menyimpan persediaan CPO. Suplai CPO diperoleh dari truk tangki, *dry port*, pabrik kelapa sawit dan pabrik pengolahan inti sawit. Sistem tangki timbun terdiri dari 3 proses, diantaranya proses *loading*, penimbunan dan *unloading*. Proses *loading* merupakan proses pengiriman CPO dari *buffer tank* menuju tangki timbun. Pada proses tersebut terjadi pengaturan aliran dan level. Proses penimbunan merupakan proses penyimpanan CPO dalam tangki, dimana kualitas CPO yang tersimpan harus dijaga dengan mengendalikan temperatur tangki agar berada pada kisaran 50°C sampai 55°C. Selain itu, pengendalian temperatur tersebut juga bertujuan agar pembacaan sensor aliran saat proses *unloading* CPO lebih akurat.

Sistem pengaturan temperatur memiliki peranan penting di industri. *Real plant* biasanya memiliki variabel yang kompleks dan *nonlinier*. Metode adaptif sebaiknya diterapkan untuk mengatasi kondisi tersebut. Pada sistem tangki penimbunan minyak kelapa sawit masih menerapkan sistem kontrol *on/off* untuk mengendalikan semua proses yang berlangsung.

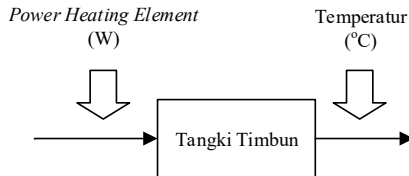
Sistem tangki timbun CPO yang dirancang pada penelitian ini berupa simulasi *plant* dan kontroler dengan metode tertentu untuk menjawab kebutuhan di industri khususnya pada penimbunan minyak kelapa sawit CPO. Metode pengendalian yang dibutuhkan harus memiliki kemampuan untuk menyesuaikan parameter kontroler terhadap perubahan yang terjadi pada *plant* saat sistem beroperasi. Berdasarkan kebutuhan tersebut, pada penelitian ini diterapkan metode PID *self-tuning* agar performa sistem memenuhi kriteria yang diinginkan.

#### **1.2 Pemodelan *Plant***

Untuk mempermudah analisa kondisi sistem dan perancangan kontroler, maka *plant* tangki timbun dan sensor yang terpasang dimodelkan berdasarkan hukum fisisnya untuk kemudian dicari fungsi alihnya. Pemodelan ini meliputi pemodelan *plant* tangki timbun dan pemodelan sensor RTD yang dijabarkan sebagai berikut.

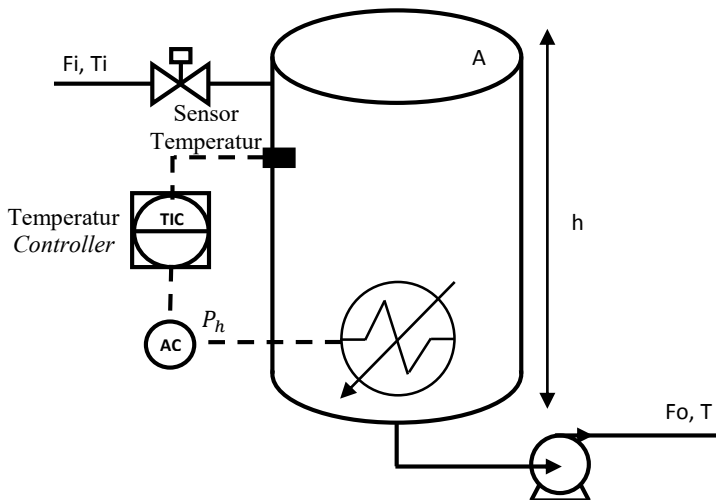
### 1.2.1 Pemodelan *Plant* Tangki Timbun

Pada proses pemodelan ini terdapat perubahan satuan dari parameter-parameter yang terlibat. Gambar 3.1 menunjukkan proses pemodelan sistem pengaturan temperatur dalam tangki.



**Gambar 2.1** Diagram pemodelan sistem pengaturan temperatur

Sistem tangki timbun yang dirancang terdiri dari *plant* tangki, sistem pemanas dengan menggunakan pemanas elektrik, dan sensor temperatur RTD. Lebih lengkap mengenai sistem tangki timbun CPO seperti pada Gambar 3.2.



**Gambar 2.2** P&ID sistem pengaturan temperatur tangki timbun

Aliran masuk CPO dengan temperatur yang bervariasi akan ditampung dalam tangki timbun. Penambahan temperatur CPO dalam tangki timbun dilakukan menggunakan pemanas elektrik. Ketika

temperatur terukur masih kurang dari 50°C, maka kontroler akan mengatur daya elemen pemanas untuk memanaskan CPO sesuai kebutuhan. Parameter-parameter yang digunakan untuk sistem pengaturan temperatur yang dirancang tercantum pada Tabel 3.1.

**Tabel 2.1** Parameter pemodelan *plant*

Simbol	Keterangan	Satuan
$F_i$	Aliran cairan yang masuk ke tangki	$\text{m}^3 / \text{jam}$
$F_o$	Aliran cairan yang keluar dari tangki	$\text{m}^3 / \text{jam}$
$T_i$	Temperatur <i>input</i>	$^{\circ}\text{C}$
$T$	Temperatur <i>output</i>	$^{\circ}\text{C}$
$h$	Tinggi level cairan (dengan asumsi cairan dalam tangki penuh dan konstan)	$\text{m}$
$A$	Luas alas tangki	$\text{m}^2$
$V$	Volume tangki	$\text{m}^3$
$m$	Massa cairan	$\text{kg}$
$\rho$	Massa jenis zat cair	$\text{kg} / \text{m}^3$
$W_{\text{in}}$	Energi masuk	$\text{J}$
$W_{\text{out}}$	Energi keluar	$\text{J}$
$C_v$	Kalor jenis zat cair dengan volume konstan	$\text{J} / \text{kg } ^{\circ}\text{C}$

Pada sistem ini model matematika *plant* diperoleh berdasarkan hukum-hukum fisisnya. Berdasarkan hukum termodinamika, kalor dalam tangki dapat dirumuskan dengan persamaan (3.1).

$$W = m.C_v.\Delta T = m.C_v.(T - T_i) \quad (3.1)$$

Berdasarkan hukum kesetimbangan energi, berlaku bahwa akumulasi energi dalam tangki sama dengan energi yang masuk dikurangi energi keluar ditambahkan dengan usaha dari luar.

$$m.C_v \cdot \frac{dT}{dt} = W_{in} - W_{out} + K_h P_h \quad (3.2)$$

Pada saat terjadi aliran cairan masuk dan keluar tangki, volume diabaikan, sehingga :

$$\frac{A.h}{Fo} \cdot \frac{dT}{dt} = \frac{F_i}{Fo} T_i - T + \frac{K_h P_h}{\rho.C_v Fo} \quad (3.3)$$

Pengaturan temperatur terjadi saat level cairan di dalam tangki penuh dan dijaga tetap / tidak berkurang. Berdasarkan kondisi tersebut diperoleh :

$$\frac{A.h}{Fo} \cdot \frac{dT}{dt} = T_i - T + \frac{K_h P_h}{\rho.C_v Fo} \quad (3.4)$$

Diasumsikan bahwa temperatur dari cairan yang masuk ke dalam tangki (**T<sub>i</sub>**) konstan, sehingga :

$$\frac{A.h}{Fo} \cdot \frac{dT}{dt} = -T + \frac{K_h P_h}{\rho.C_v Fo} \quad (3.5)$$

Transformasi *Laplace* dari persamaan (3.5) adalah sebagai berikut :

$$\frac{A.h}{Fo} .s.T(s) + T(s) = \frac{K_h}{\rho.C_v Fo} .P_h(s) \quad (3.6)$$

$$\frac{T(s)}{P_h(s)} = \frac{\frac{K_h}{\rho.C_v.Fo}}{\left(\frac{V}{Fo}\right)s + 1} \quad (3.7)$$

Beberapa parameter diketahui untuk mendapatkan model matematika *plant* berdasarkan hukum fisis. Parameter-parameter yang digunakan dan diketahui nilainya tercantum pada Tabel 3.2.

**Tabel 2.2** Parameter *plant* tangki timbun

Simbol	Keterangan	Nilai	Satuan
Fo	Aliran air keluar	50	m <sup>3</sup> / jam
Cv	Kapasitas panas air dengan volume konstan	4.18	J / g °C
$\rho$	Massa jenis zat cair	1	g / m <sup>3</sup>
V	Volume tangki	5251.137	m <sup>3</sup>
$P_h$	<i>Maximum power heating element</i>	840	W
$K_h$	<i>Heating element power gain</i>	504	J

Dari Tabel 3.2 dapat diperoleh fungsi alih tangki timbun seperti pada persamaan (3.8).

$$\frac{T(s)}{P_h(s)} = \frac{2.411}{105.022s + 1} \quad (3.8)$$

### 1.2.2 Pemodelan Sensor RTD

Tipe sensor temperatur yang akan digunakan pada sistem ini adalah RTD dengan rentang masukan temperatur 0° sampai 55° C. Keluaran yang dihasilkan berupa resistansi yang akan dikonversi menjadi tegangan antara 0 – 10 Volt. Pada pemodelan ini diperoleh penguatan RTD yang dilambangkan dengan  $K_{RTD}$ , dimana nilai penguatan tersebut dirumuskan sebagai perbandingan antara tegangan keluaran dengan temperatur cairan dalam tangki.

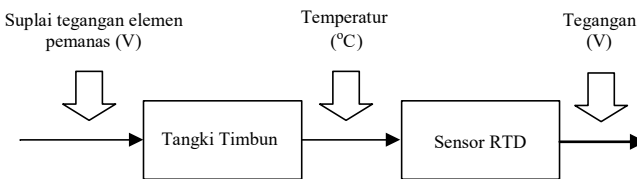
$$K_{RTD} = \frac{\text{Tegangan output (Volt)}}{\text{Temperatur (°C)}} \quad (3.9)$$

Dari pemodelan di atas diperoleh *gain* sensor yang merupakan persamaan masukan dan keluarannya seperti pada persamaan (3.12).

$$K_{RTD} = \frac{10 \text{ (Volt)}}{55 \text{ (}^{\circ}\text{C)}} \quad (3.10)$$

### 1.2.3 Fungsi Alih Keseluruhan

Sistem secara keseluruhan terdiri dari plant tangki dan pemanas elektriknya serta sensor temperatur yang diunakan untuk mendeteksi temperatur dalam tangki. Diperoleh diagram blok sistem secara keseluruhan seperti pada Gambar 3.3 dengan fungsi alihnya pada persamaan (3.11).

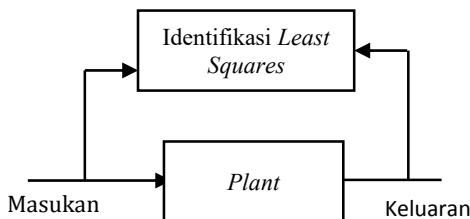


**Gambar 2.3** Diagram blok sistem

$$\frac{Y(s)}{X(s)} = \frac{13.260}{105.022s + 1} \quad (3.11)$$

### 1.3 Identifikasi *Plant*

Identifikasi dilakukan dengan pendekatan hasil model orde pertama, orde 2 dan orde 3 model. Dari ketiga model hasil identifikasi tersebut kemudian akan dicari yang paling valid untuk digunakan pada proses selanjutnya yaitu perancangan kontroler. Ada 2 jenis sinyal masukan yang digunakan yaitu *unit step* dan PRBS. Diagram blok proses identifikasi adalah seperti pada Gambar 3.4.



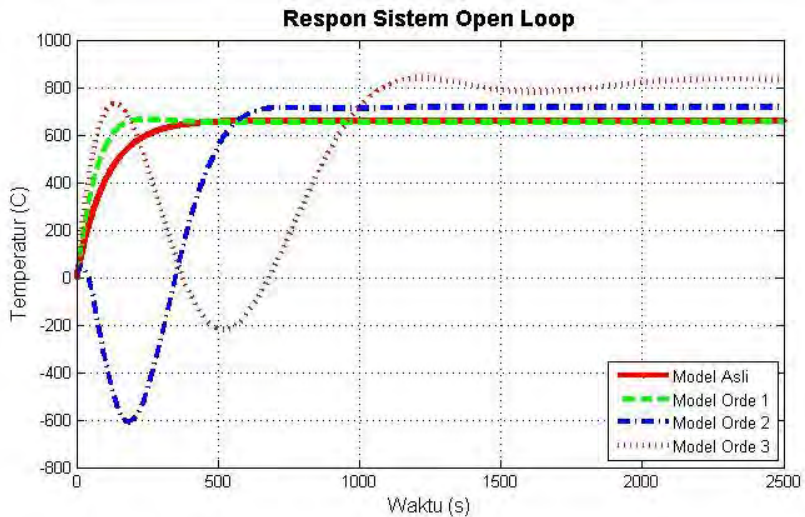
**Gambar 2.4** Diagram blok proses identifikasi



Dengan masukan *unit step* dan PRBS diperoleh model orde 1, orde 2 dan orde 3 seperti pada Tabel 2.3 dan Tabel 2.4.

### Model Hasil Identifikasi dengan Masukan *Unit Step*

Plot respon sistem *open loop* dengan model hasil identifikasi yang diberikan masukan *unit step* memiliki perbandingan respon seperti pada Gambar 2.5. Masukan yang diberikan pada sistem *open loop* yang diuji adalah berupa sinyal *unit step*.



**Gambar 2.5** Hasil identifikasi dengan masukan *unit step*

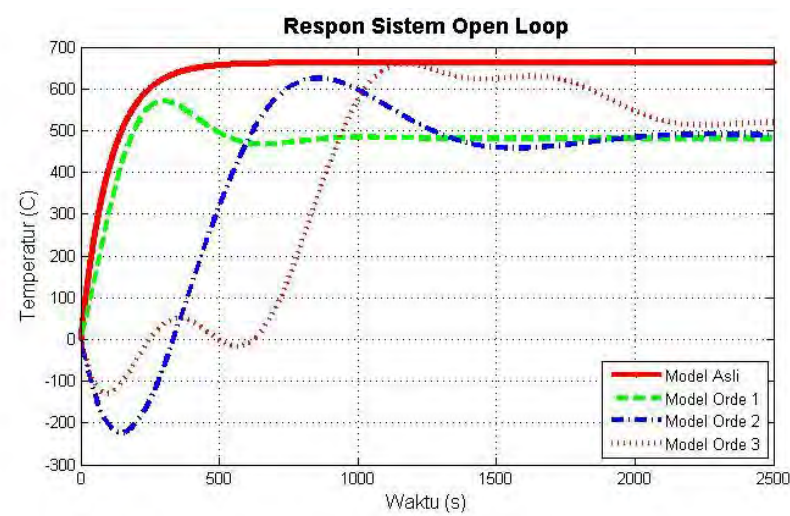
Kesalahan model hasil identifikasi terhadap model sistem asli diperhitungkan dengan metode validasi untuk mendapatkan model yang paling valid, yaitu model dengan nilai kesalahan terkecil. Metode validasi yang paling sesuai untuk respon seperti pada Gambar 2.5 adalah *Integral of Squared Error (ISE)* dikarenakan respon model tersebut memiliki selisih yang besar pada bagian peralihan. Fungsi alih model hasil identifikasi dengan pendekatan orde 1, orde 2 dan orde 3 seperti pada Tabel 2.3.

**Tabel 2.3** Hasil identifikasi dengan masukan *unit step*

Model Plant	
Orde 1	$\frac{Y(z)}{X(z)} = \frac{13.1652}{z + 0.00715}$
Orde 2	$\frac{Y(z)}{X(z)} = \frac{13.3027z}{z^2 + 0.00209z - 0.00531}$
Orde 3	$\frac{Y(z)}{X(z)} = \frac{13.9375z^2}{z^3 - 0.00014z^2 + 0.00976z - 0.05469}$

**Model Hasil Identifikasi dengan Masukan PRBS**

Plot respon sistem dengan model hasil identifikasi yang diberikan masukan PRBS memiliki perbandingan respon seperti pada Gambar 2.6.



**Gambar 2.6** Hasil identifikasi dengan masukan PRBS

Kesalahan pada respon tersebar dari daerah peralihan sampai keadaan tunak. Untuk mendapatkan model yang paling valid perlu dilakukan

validasi. Ada beberapa metode validasi, yang paling sesuai dengan kondisi seperti pada Gambar 2.6 adalah metode *Root Mean Square Error* (RMSE) untuk model yang memiliki kesalahan tersebar di seluruh daerah respon. model yang paling valid dipilih berdasarkan nilai kesalahan terkecil dari 3 model hasil identifikasi. Hasil identifikasi dengan pendekatan model orde pertama, orde 2 dan orde 3 seperti pada Tabel 2.4.

**Tabel 2.4** Hasil identifikasi dengan masukan PRBS

<b>Model Plant</b>	
<b>Orde 1</b>	$\frac{Y(z)}{X(z)} = \frac{11.1422}{z + 0.1566}$
<b>Orde 2</b>	$\frac{Y(z)}{X(z)} = \frac{10.75z}{z^2 + 0.0168z + 0.1931}$
<b>Orde 3</b>	$\frac{Y(z)}{X(z)} = \frac{10.8003z^2}{z^3 + 0.0116z^2 + 0.0113z + 0.1719}$

## 1.4 Validasi Model

Berdasarkan respon pada Gambar 2.5 dan Gambar 2.6 dipilih dua metode validasi dengan kriteria kesalahan yang sesuai yaitu ISE dan RMSE. Data hasil validasi model dengan masukan identifikasi *unit step* disajikan pada Tabel 2.5.

**Tabel 2.5** Validasi model dengan masukan identifikasi *unit step*

<b>Model Plant</b>		<b>ISE</b>
<b>Orde 1</b>	$\frac{Y(z)}{X(z)} = \frac{13.1652}{z + 0.00715}$	3.3687x10 <sup>7</sup>
<b>Orde 2</b>	$\frac{Y(z)}{X(z)} = \frac{13.3027z}{z^2 + 0.00209z - 0.00531}$	2.9446x10 <sup>9</sup>
<b>Orde 3</b>	$\frac{Y(z)}{X(z)} = \frac{13.9375z^2}{z^3 - 0.00014z^2 + 0.00976z - 0.05469}$	3.3345x10 <sup>9</sup>

Berdasarkan Tabel 2.5 diperoleh model yang paling valid dengan nilai kesalahan terkecil yaitu model orde pertama. Oleh karena itu, dipilih model tersebut untuk diaplikasikan pada sistem ini. Alasan lainnya, model orde pertama dipilih untuk diaplikasikan karena model tersebut lebih sederhana, sehingga lebih memungkinkan untuk diaplikasikan pada sistem *real*.

Data hasil validasi model dengan masukan identifikasi PRBS disajikan pada Tabel 2.6.

**Tabel 2.6** Validasi model dengan masukan identifikasi PRBS

Model <i>Plant</i>		RMSE
Orde 1	$\frac{Y(z)}{X(z)} = \frac{11.1422}{z + 0.1566}$	146.2643
Orde 2	$\frac{Y(z)}{X(z)} = \frac{10.75z}{z^2 + 0.0168z + 0.1931}$	231.2895
Orde 3	$\frac{Y(z)}{X(z)} = \frac{10.8003z^2}{z^3 + 0.0116z^2 + 0.0113z + 0.1719}$	252.4001

Berdasarkan Tabel 2.6 diperoleh model yang paling valid dengan nilai kesalahan terkecil yaitu model orde pertama, seperti hasil validasi dengan masukan *unit step*. Oleh karena itu, dipilih model tersebut untuk diaplikasikan sebagai pendekatan dalam proses estimasi.

### 1.5 Perancangan Kontroler

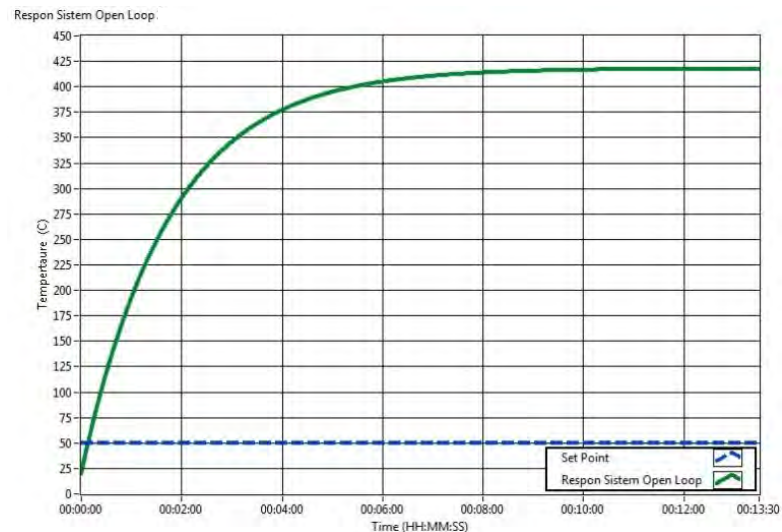
Kontroler dirancang untuk mengatasi permasalahan pada sistem ini yaitu perubahan parameter *plant* yang tidak dapat diprediksi. Perancangan tersebut meliputi kontroler PID. Akan tetapi, karena *plant* dari hasil pemodelan berupa model orde pertama, penggunaan kontroler PI sudah cukup untuk memperoleh respon sistem dengan performansi sesuai dengan spesifikasi yang diinginkan. Setelah merancang kontroler PI selanjutnya adalah proses estimasi parameter dan proses *update* parameter kontroler PI menggunakan metode *self-tuning regulator*.

### 1.5.1 Perancangan Kontroler PI

Perancangan kontroler dilakukan berdasarkan kriteria yang ditentukan dan respon sistem *open loop* yang diperoleh, yaitu respon *plant* hasil pemodelan berdasarkan hukum fisis untuk merancang kontroler PI yang sesuai dengan spesifikasi yang diinginkan.

#### Sistem *Open Loop*

Fungsi alih pengaturan temperatur cairan dalam tangki yang diterapkan untuk tangki timbun CPO, seperti pada persamaan (3.13) diuji dengan masukan *unit step*, sehingga diperoleh respon seperti pada Gambar 3.7.

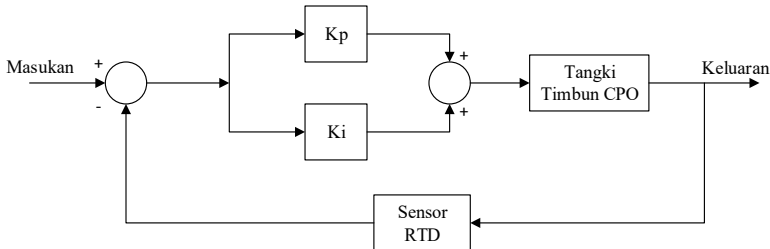


**Gambar 2.7** Respon sistem *open loop*

Sebelum temperatur diatur, dalam tangki CPO diasumsikan terdapat temperatur ruang sebesar 20°C, sehingga respon sistem tidak mulai dari 0°C tetapi dari nilai *offset* yang diatur yaitu 20°C.

Berdasarkan hasil pengujian sistem *open loop* pada Gambar 3.7, terjadi kesalahan keadaan tunak yang besar antara respon *plant* dengan *set point* yang diberikan. Sistem ini diperbaiki dengan merancang kontroler PI.

Spesifikasi perancangan kontroler PI yaitu tidak ada kesalahan keadaan tunak dan waktu pemanasan untuk mencapai temperatur maksimal yang awalnya hampir memakan waktu 9 jam yang dalam simulasi ini dipercepat dalam satuan menit menjadi 9 menit. Waktu tersebut dibuat 3 kali lebih cepat menjadi 3 menit. Diagram blok sistem dengan kontroler PI seperti pada Gambar 2.8.



**Gambar 2.8** Diagram blok sistem dengan kontroler PI

Perancangan kontroler PI dilakukan menggunakan metode analitik, dengan prosedur di bawah ini.

### Perancangan Kontroler Proporsional

Perancangan kontroler P dengan spesifikasi sistem yang diinginkan adalah respon 3 kali lebih cepat dari respon sistem *open loop* dimana persamaan sistem seperti pada persamaan (3.12).

$$\frac{Y(s)}{X(s)} = \frac{13.260}{105.022s + 1} \quad (3.12)$$

Berdasarkan spesifikasi respon sistem yang diinginkan, diperoleh  $\tau^* = 35.007$ , sehingga diperoleh *gain* proporsional  $K_p = 0.15083$ .

### Perancangan Kontroler Proporsional dan Integrator

Perancangan kontroler proporsional hanya untuk mempercepat respon sistem, bukan untuk menghilangkan kesalahan keadaan tunak, sehingga harus ditambahkan integrator dengan tujuan untuk menghilangkan kesalahan keadaan tunak, sehingga sistem dapat mencapai *set point* yang diinginkan. Penambahan integrator dipilih

$\tau_i = \tau = 105.022$ . Penentuan tersebut bertujuan agar respon hasil perancangan kontroler PI berupa respon sistem orde pertama.

### 1.5.2 Perancangan Metode Estimasi Parameter

Proses estimasi dilakukan untuk memprediksikan parameter-parameter *plant* untuk merancang kontroler baru agar dapat menyesuaikan dengan perubahan yang terjadi pada *plant* saat beroperasi. Metode yang diimplementasikan pada penelitian ini adalah metode *Least Squares Estimation* (LSE). Proses estimasi ini menggunakan masukan sinyal kontrol dan sinyal keluaran *plant*. Dengan menggunakan proses perhitungan matematis, diperoleh parameter *plant* dari kedua sinyal tersebut sesuai orde yang ditentukan. Proses perhitungan matematis yang telah dijabarkan sebelumnya pada Sub Bab **Error! Reference source not found.** diterjemahkan dalam program agar proses estimasi berlangsung dengan cepat dan *online*, artinya program dapat terus-menerus berjalan untuk melakukan estimasi sehingga setiap perubahan yang terjadi pada *plant* dapat diprediksi.

Model hasil estimasi diharapkan sesuai dengan model asli. Jadi diharapkan kesalahan hasil estimasi adalah 0, sehingga :

$$Y = \varphi\theta$$

Berdasarkan nilai kuadrat kesalahan terkecil, diperoleh parameter plant hasil estimasi dari persamaan matematis berikut :

$$\theta = (\varphi^T \varphi)^{-1} \varphi^T Y$$

Pembahasan perancangan metode LSE di atas adalah untuk model orde pertama. Model orde 2 dan seterusnya harus disesuaikan bentuk persamaan model yang dihasilkan dan parameter-parameternya.

### 1.5.3 Perancangan Metode *Self-Tuning Regulator*

Metode *self-tuning regulator* terdiri dari 2 *loop*, yang pertama merupakan sistem *closed loop* pada umumnya yang meliputi kontroler, *plant* dan sinyal *feedback*, sedangkan *loop* lainnya melibatkan proses estimasi parameter, perancangan kontroler baru dan proses *update* parameter kontroler. Proses estimasi parameter menggunakan sinyal kontrol yang masuk ke *plant* dan sinyal keluaran *plant*, melalui proses

estimasi tersebut dihasilkan parameter *plant* untuk merancang kontroler agar dapat melakukan adaptasi terhadap perubahan-perubahan yang terjadi pada *plant*. Prinsip kerja metode *self-tuning regulator* secara rinci seperti pada **Error! Reference source not found.**

Hasil perancangan kontroler baru dengan metode tersebut digunakan untuk *update* parameter PI yang digunakan dari awal. Dengan perubahan parameter kontroler tersebut, performa sistem dapat tetap maksimal meskipun parameter *plant* berubah. Proses perancangan kontroler baru dari model hasil estimasi dijabarkan pada Sub Bab **Error! Reference source not found.** Proses perancangan tersebut diterjemahkan dalam program agar dapat berlangsung dengan cepat dan *online*, artinya proses perancangan kontroler baru dapat dilakukan terus-menerus saat sistem sedang beroperasi dan parameter *plant* mengalami perubahan. Pada proses perancangan kontroler tersebut diperoleh nilai  $\tau$  dan  $K$  yang digunakan untuk memperoleh parameter  $K_p$  dan  $K_i$  untuk *update* parameter PI yang sebelumnya, sehingga kontroler dapat terus menyesuaikan terhadap perubahan *plant*. Nilai  $\tau$  dan  $K$  untuk model orde pertama diperoleh sebagai berikut.

$$\tau = -\frac{a_1 + 1}{2(a_1 - 1)}T_s$$

$$K = (2\tau + T_s)\frac{b_0}{T_s}$$

## 1.6 Perancangan Simulasi *Hardware in the Loop*

Berdasarkan kondisi *real* di lapangan, letak *plant* dan *control room* terpisah sejauh 15 meter. Implementasi pada penelitian ini, *plant* dibuat *virtual* pada sebuah komputer. Kontroler dianalogikan sebagai *control room* yang juga diimplementasikan pada sebuah komputer. Komputer *plant* dan kontroler dihubungkan melalui *Digital to Analog Converter* (DAC) dan *Analog to Digital Converter* (ADC) menggunakan DAQ Navi *Advantech* USB-4716. Antara komputer *virtual plant* maupun komputer kontroler dihubungkan dengan DAQ Navi *Advantech* USB-4716 menggunakan media kabel, begitu juga dengan *Port analog output* (Ao) dan *analog input* (Ai) yang juga dihubungkan dengan media kabel. Lebih detailnya mengenai komponen-komponen sistem yang akan dibuat pada





tab yaitu tab untuk menampilkan respon sistem dan tab untuk menampilkan proses estimasi parameter, meliputi grafik proses estimasi yang menunjukkan perubahan nilai parameter *plant* hasil estimasi, *digital display* yang akan menampilkan parameter-parameter hasil estimasi dan parameter kontroler baru yang sudah diperbarui sesuai dengan perubahan parameter *plant* yang terjadi selama proses di industri berjalan.

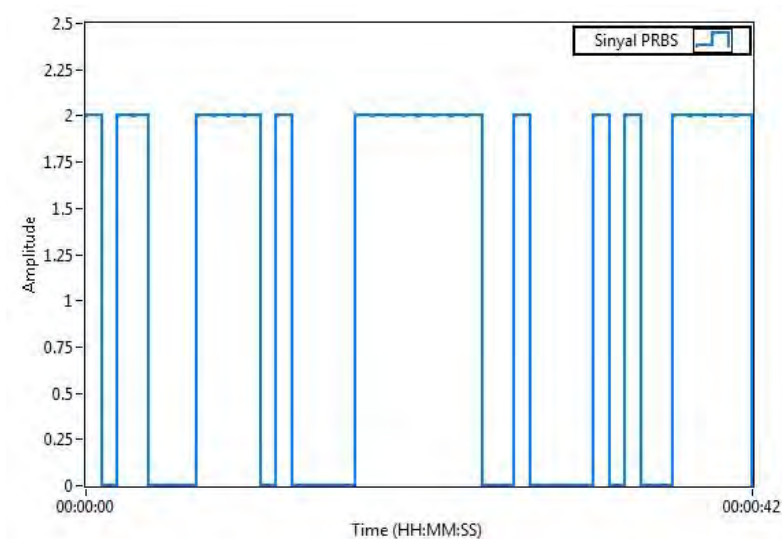
HMI *virtual plant* memiliki fitur untuk menampilkan respon sistem saja tanpa menampilkan proses estimasi, *update* parameter dan fitur-fitur lain yang pada umumnya hanya terdapat pada komputer kontroler.

## BAB 4

### PENGUJIAN DAN ANALISA

#### 2.1 Pengujian Program untuk Membangkitkan Sinyal PRBS

PRBS dibangkitkan dengan pergeseran bit dan logika XOR. Caranya dengan menghubungkan keluaran dari deretan *shift register* ke gerbang logika XOR, kemudian hasilnya dikembalikan ke masukan *shift register*. Sinyal PRBS yang dihasilkan seperti pada Gambar 2.1.



**Gambar 2.1** Sinyal PRBS

Sinyal PRBS seperti pada Gambar 2.1 diatur amplitudo maksimal 2 sedangkan minimalnya 0. Pengaturan tersebut diberikan untuk menggambarkan kondisi di industri bahwa suplai tegangan untuk elemen pemanas bernilai positif bukan negatif. PRBS dibutuhkan pada penelitian ini untuk melakukan identifikasi *plant*. Tujuannya adalah dengan masukan PRBS dapat menggambarkan proses *real* di industri yang sinyal

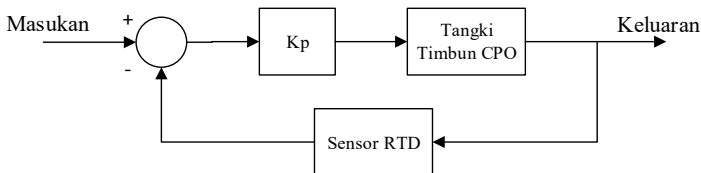
masukannya berubah-ubah, artinya ketika diberikan tegangan masukan 5 Volt, maka tegangan yang sebenarnya diterima *plant* tidak tepat 5 Volt. Akan tetapi, bisa kurang atau lebih.

## 2.2 Pengujian Sistem dengan Kontroler PI

Perancangan kontroler PI untuk sistem telah dibahas pada Bab 3 dengan spesifikasi yang telah ditentukan. Pengujian sistem dengan kontroler PI tersebut dilakukan secara bertahap yaitu pertama dengan penggunaan kontroler proporsional. Dikarenakan sistem dengan kontroler proporsional belum memenuhi performansi sistem yang ditentukan maka ditambahkan integrator. Dengan penambahan integrator tersebut sistem sudah dapat memenuhi spesifikasi perancangan yang ditentukan, sehingga tidak perlu ditambahkan fungsi derivatif. Pengujian hasil penambahan kontroler P dan PI pada sistem dibahas di Sub Bab ini.

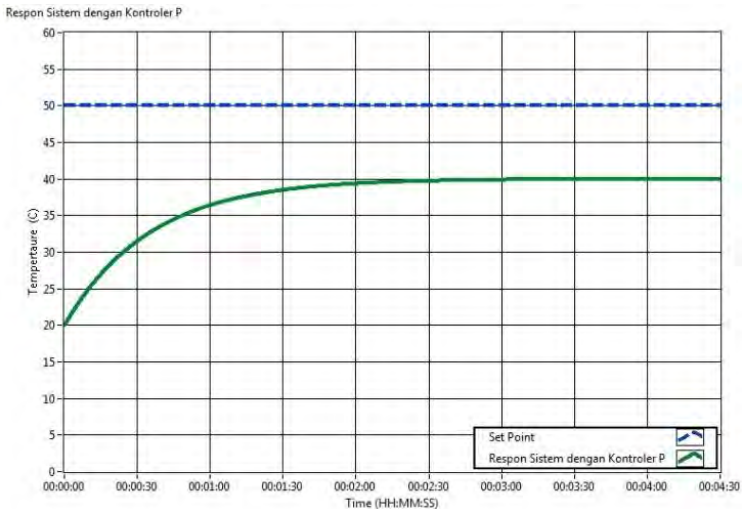
### 2.2.1 Pengujian Sistem dengan Kontroler Proporsional

Perancangan kontroler proporsional bertujuan agar respon sistem dengan kontroler menjadi 3 kali lebih cepat dari respon sistem tanpa kontroler. Berdasarkan spesifikasi tersebut, dipilih nilai *gain* proporsional yang sesuai, sehingga diperoleh  $K_p = 0.15083$ . Untuk membuktikan bahwa metode perancangan yang diterapkan benar dan nilai *gain* proporsional yang diberikan sudah sesuai, maka dilakukan pengujian sistem ditambahkan kontroler proporsional yang telah dirancang tersebut dengan *setting* seperti pada Gambar 2.2.



**Gambar 2.2** Diagram blok sistem dengan kontroler proporsional

Respon sistem dengan penambahan kontroler proporsional dapat dilihat pada Gambar 2.3.



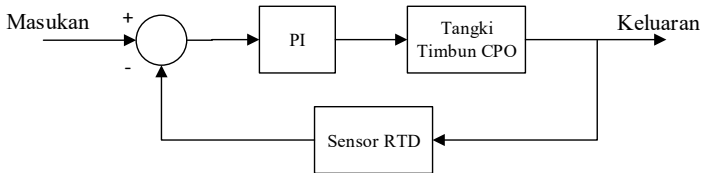
**Gambar 2.3** Respon sistem dengan kontroler proporsional

Dari Gambar 2.3 diperoleh keadaan tunak sistem dengan *set point* 50°C berada pada temperatur 40°C yang dicapai dalam waktu 2.5 menit, sedangkan berdasarkan pada Gambar 3.7 respon sistem *open loop* dapat mencapai keadaan tunak dalam waktu 8 menit. Perbandingan waktu pencapaian keadaan tunak diperoleh respon sistem dengan kontroler proporsional dapat mencapai keadaan tunak kurang lebih 3 kali lebih cepat daripada respon sistem *open loop*. Berdasarkan hasil pengujian tersebut, spesifikasi perancangan kontroler proporsional terpenuhi. Akan tetapi, respon memiliki kesalahan keadaan tunak sebesar 20%, sehingga sistem tidak dapat mencapai *set point* yang diberikan. Kesalahan tersebut diatasi dengan penambahan integrator pada sistem, sehingga kontroler yang diterapkan adalah PI.

### 2.2.2 Pengujian Sistem dengan Kontroler Proporsional Integrator

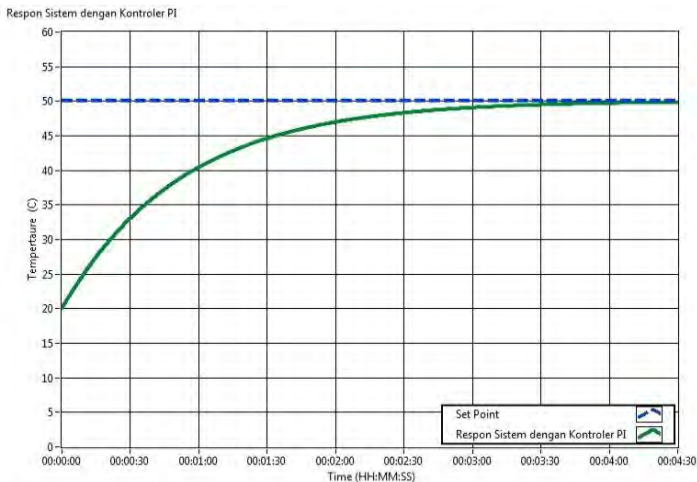
Perancangan kontroler PI bertujuan untuk menghilangkan kesalahan keadaan tunak yang terjadi saat sistem hanya diberikan kontroler proporsional. Spesifikasi perancangan kontroler tersebut selain menghilangkan kesalahan keadaan tunak adalah agar respon yang dihasilkan berupa respon model orde pertama, sehingga dipilih

$\tau_i = \tau = 105.022$ . *Setting* pengujian sistem dengan kontroler PI adalah seperti pada Gambar 2.4.



**Gambar 2.4** Diagram blok sistem dengan kontroler PI

Respon sistem dengan penambahan integrator seperti *setting* pengujian Gambar 2.4 dapat dilihat pada Gambar 2.5.



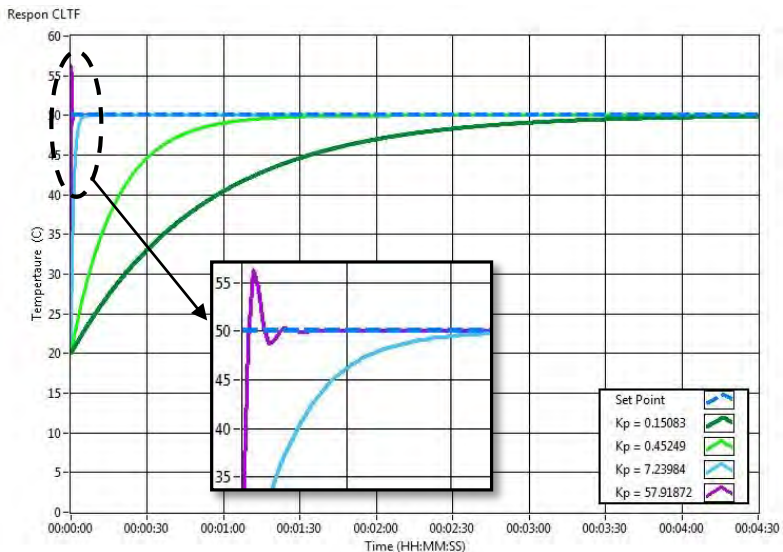
**Gambar 2.5** Respon sistem dengan kontroler PI

Dari Gambar 2.5 diperoleh, dengan penambahan integrator, kesalahan keadaan tunak sistem dapat mencapai 0%. Sistem mencapai keadaan tunak sesuai *set point* yang diberikan, dalam waktu kurang lebih 3 menit. Waktu yang dibutuhkan sistem tersebut untuk mencapai keadaan tunak sama dengan waktu yang dibutuhkan sistem dengan kontroler proporsional. Perbedaan respon kedua sistem tersebut terletak pada nilai

keadaan tunak yang dicapai, sehingga dapat disimpulkan bahwa respon sistem dengan penambahan integrator dapat memenuhi spesifikasi yang ditentukan.

### 2.2.3 Pengujian Sistem dengan Kontroler PI untuk Variasi $K_p$

*Setting* pengujian sistem dengan kontroler PI menggunakan variasi  $K_p$  sama seperti Gambar 2.4. Pemberian variasi nilai tersebut bertujuan untuk mengamati pengaruh besar kecilnya nilai  $K_p$  terhadap respon sistem. Hasil pengujian dapat diamati pada Gambar 2.6.

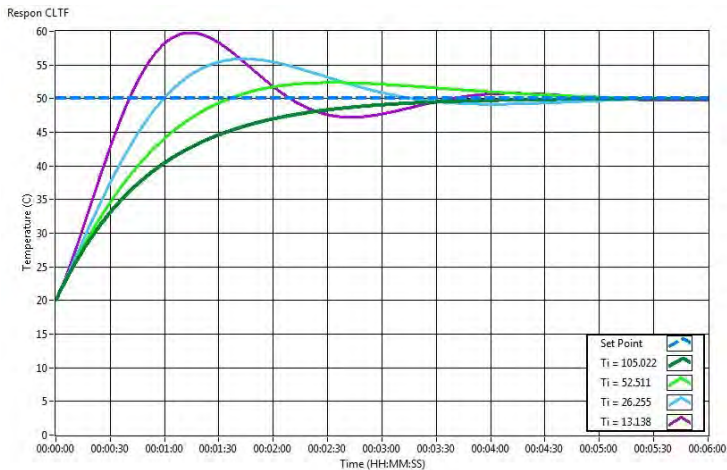


**Gambar 2.6** Respon sistem dengan kontroler PI untuk variasi  $K_p$

Berdasarkan hasil pengujian pada Gambar 2.6 terbukti bahwa nilai  $K_p$  mempengaruhi cepat atau lambatnya respon sistem. Semakin besar nilai *gain* proposional yang diberikan, respon sistem akan semakin cepat. Akan tetapi, apabila nilai yang diberikan terlalu besar, maka sistem akan berosilasi. Seperti yang terlihat pada Gambar 2.6, dengan nilai  $K_p$  yang paling besar sistem mengalami *overshoot* dan osilasi di bagian awal. Akan tetapi responnya lebih cepat dari yang lain.

#### 2.2.4 Pengujian Sistem dengan Kontroler PI untuk Variasi $\tau_i$

*Setting* pengujian sistem dengan kontroler PI menggunakan variasi  $\tau_i$  sama seperti Gambar 4.4. Pemberian variasi nilai tersebut bertujuan mengamati pengaruh besar kecilnya nilai  $\tau_i$  terhadap respon sistem untuk menghilangkan kesalahan keadaan tunak. Hasil pengujian dapat diamati pada Gambar 2.7.



**Gambar 2.7** Respon sistem dengan kontroler PI untuk variasi  $\tau_i$

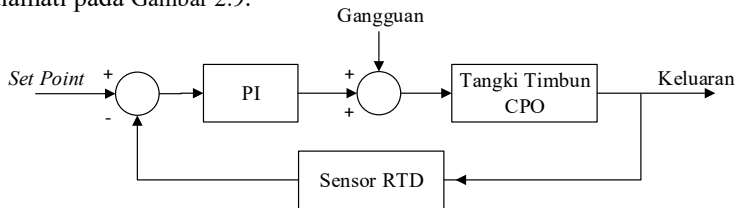
Nilai  $\tau_i$  pada awalnya diberikan sama dengan  $\tau$  sehingga respon sistem menyerupai orde pertama. Perubahan nilai  $\tau_i$  mempengaruhi perubahan denominator menjadi orde 2. Dengan perubahan tersebut, berarti persamaan karakteristik sistem juga berubah menjadi orde 2, begitu juga dengan responnya. Seperti pada Gambar 2.7 dapat diamati bahwa ketika  $\tau_i$  tidak sama dengan  $\tau$  yaitu 105.022, respon sistem mengalami *overshoot* sebelum menuju *set point*, seperti respon orde 2.

#### 2.2.5 Pengujian Sistem dengan Kontroler PI Ditambahkan Gangguan

*Setting* pengujian sistem dengan kontroler PI yang diberikan gangguan seperti Gambar 2.8. Pemberian gangguan tersebut bertujuan

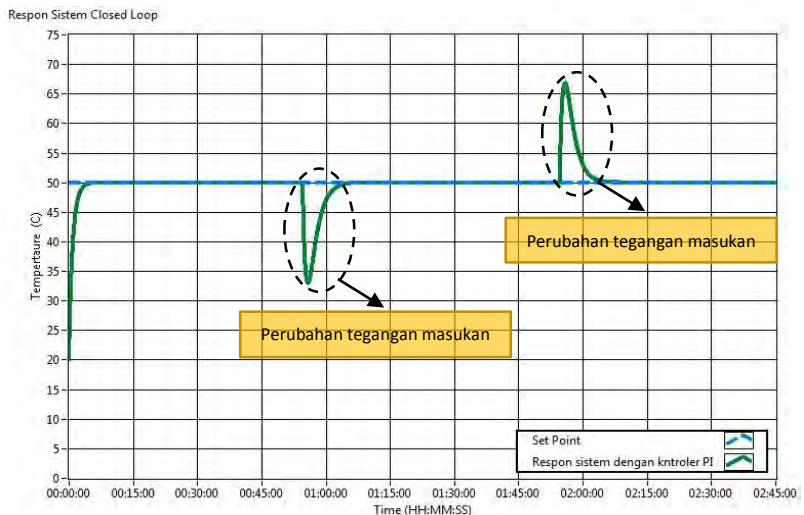


untuk menguji ketahanan sistem saat ada masukan yang muncul secara tidak terduga. Masukan tersebut dapat berupa perubahan tegangan atau *gain* pemanas elektrik dan aliran cairan keluar. Hasil pengujian dapat diamati pada Gambar 2.9.



**Gambar 2.8** Diagram blok sistem dengan kontroler PI ditambahkan gangguan

Respon sistem dengan kontroler PI yang diberikan gangguan seperti *setting* pengujian Gambar 2.8 dapat dilihat pada Gambar 2.9.

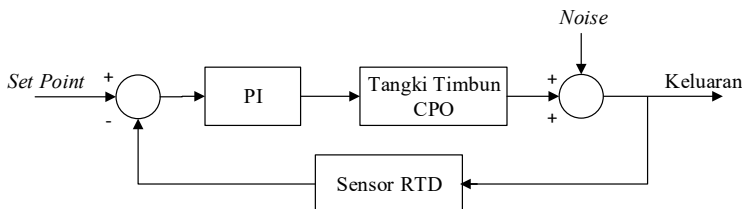


**Gambar 2.9** Respon sistem dengan kontroler PI ditambahkan gangguan

Pemberian gangguan pada sistem dengan kontroler PI bertujuan untuk menguji ketahanan sistem terhadap masukan yang muncul secara tiba-tiba, seperti perubahan tegangan untuk pemanas elektrik, perubahan aliran cairan keluar dari tangki atau variabel masukan lain yang dapat berubah-ubah. Pada pengujian ini, diberikan perubahan nilai suplai tegangan pemanas elektrik yang dapat mempengaruhi keluaran sistem. Hasilnya saat timbul gangguan pada sistem, respon mengalami *overshoot* atau *undershoot*. Akan tetapi, dengan penggunaan kontroler PI keluaran sistem dapat kembali sesuai dengan *set point* yang diberikan, hal ini ditunjukkan dengan respon yang mampu kembali *steady* seperti saat tanpa gangguan. Respon tersebut dapat diamati pada Gambar 2.9.

### 2.2.6 Pengujian Sistem dengan Kontroler PI Ditambahkan *Noise*

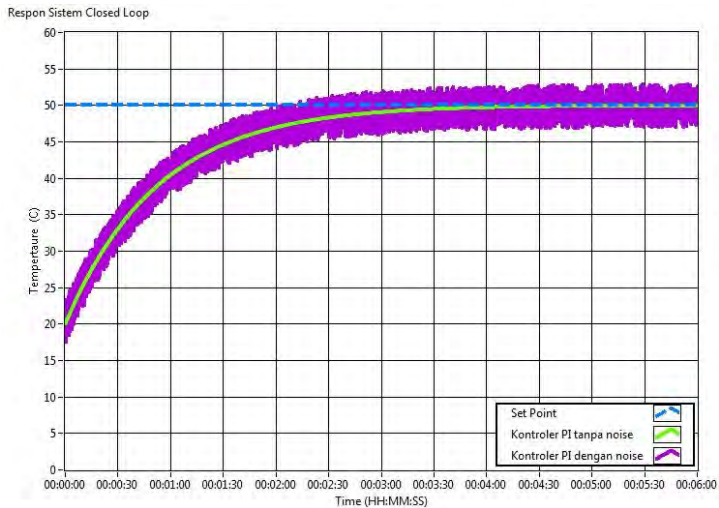
Sistem dengan kontroler PI diberikan *noise* agar merepresentasikan sistem *real* di industri yang memiliki kesalahan pengukuran, kesalahan tersebut dapat mengakibatkan kesalahan pembacaan sensor. Pengujian ini bertujuan untuk mengukur seberapa sensitif sistem dengan kontroler PI terhadap perubahan kecil. *Setting* pengujian sistem seperti Gambar 4.10.



**Gambar 2.10** Diagram blok sistem dengan kontroler PI ditambahkan *noise*

Respon hasil pengujian sistem dengan kontroler PI yang diberikan *noise* seperti *setting* pengujian Gambar 4.10 dapat dilihat pada Gambar 2.11. Pada gambar tersebut diperoleh bentuk respon sistem dengan *noise* menyerupai respon sistem tanpa diberikan *noise*. Akan tetapi, respon sistem dengan *noise* tersebut memiliki *range* lebih lebar dibandingkan sinyal respon sistem tanpa *noise*. Hal tersebut terjadi karena nilai keluaran yang dihasilkan sistem dengan *noise* tidak tepat seperti nilai yang diharapkan, melainkan mengalami kekurangan atau kelebihan nilai. Karena ketidak tepatan tersebutlah yang menyebabkan *range* keluaran sistem dengan *noise* lebih lebar, tidak seperti sistem tanpa *noise* yang

hanya digambarkan dengan sebuah garis tipis yang berarti *range* keluarannya kecil.



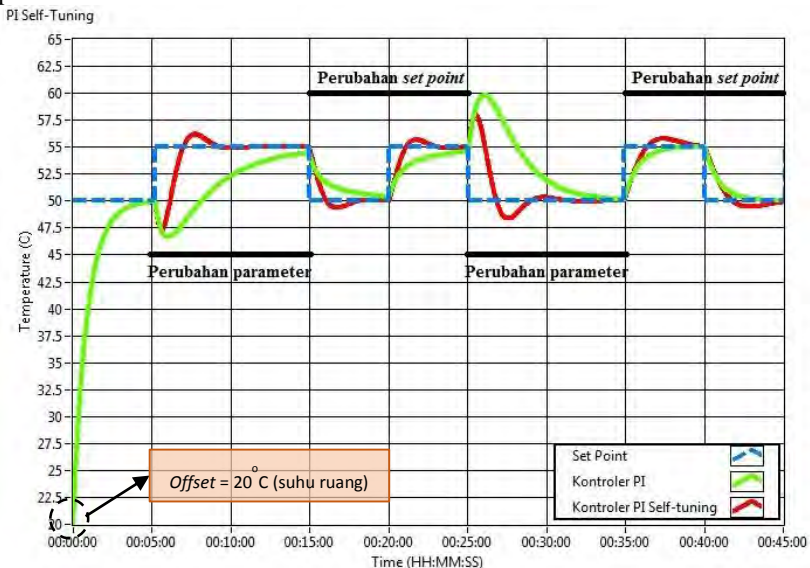
**Gambar 2.11** Respon sistem dengan kontroler PI ditambahkan *noise*

## 2.3 Pengujian Sistem dengan Kontroler PI *Self-Tuning*

Selama sistem *real* beroperasi, *plant* akan terus-menerus mengalami perubahan parameter tanpa dapat diprediksi. Oleh karena itu, sistem yang awalnya hanya menggunakan kontroler PI kemudian ditambahkan mekanisme *update* parameter agar mampu menyesuaikan dengan setiap perubahan parameter yang terjadi pada *plant*. Parameter kontroler dapat berubah-ubah sesuai dengan proses estimasi yang telah berlangsung..

Dengan metode *self-tuning*, perubahan parameter yang terjadi pada *plant* dapat diprediksikan melalui proses estimasi. Kontroler baru untuk proses *update* parameter PI dirancang berdasarkan hasil estimasi parameter agar dapat menyesuaikan dengan perubahan yang terjadi pada *plant* saat beroperasi. Proses estimasi ini menggunakan masukan sinyal kontrol dan sinyal keluaran *plant*. Dengan menggunakan proses perhitungan matematis, diperoleh parameter *plant* dari kedua sinyal tersebut sesuai orde yang ditentukan. *Setting* pengujian sistem dengan kontroler PI *self-tuning* seperti pada **Error! Reference source not**

**found..** Respon sistem dengan PI *self-tuning* yang diperoleh dapat dilihat pada Gambar 2.12.



**Gambar 2.12** Respon sistem dengan kontroler PI *Self-Tuning*

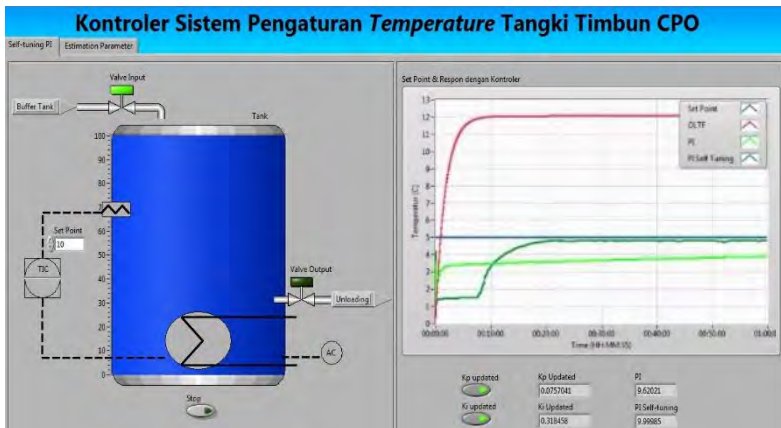
Berdasarkan respon seperti pada Gambar 2.12 diperoleh bahwa respon sistem dengan kontroler PI *self-tuning* lebih cepat daripada sistem dengan kontroler PI konvensional. Saat terjadi perubahan parameter *plant*, *plant* baru akan diestimasi sehingga diperoleh parameter kontroler baru untuk *update*. Pada respon terjadi osilasi, hal tersebut menyatakan bahwa sistem masih mencari parameter PI yang sesuai, selanjutnya ketika diperoleh parameter PI yang sesuai maka respon sistem akan mencapai keadaan tunak.

## 2.4 Simulasi *Hardware in the Loop*

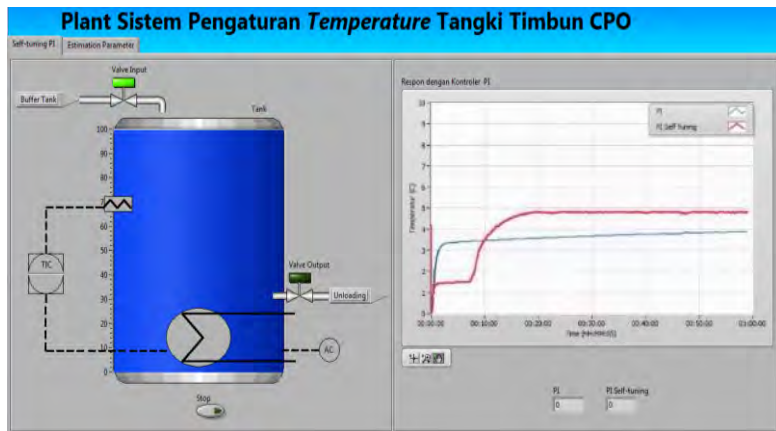
Sistem yang dirancang diuji pada satu komputer terlebih dahulu. Ketika performanya sesuai dengan yang diinginkan, maka sistem tersebut diimplementasikan di dua komputer. Metode kontrol dirancang pada komputer yang berperan sebagai kontroler, sedangkan *plant* dan sensor dirancang pada komputer yang berperan sebagai *virtual plant*. Kedua komputer tersebut dihubungkan melalui data akuisisi yang berperan

sebagai ADC/DAC dengan *setting* seperti pada Konfigurasi perangkat keras yang digunakan, diterapkan pada sistem ini seperti diagram blok Gambar 3.9.

Hasil pengujian sistem yang diimplementasikan pada 2 komputer tertera pada Gambar 2.13 untuk komputer kontroler dan Gambar 2.14 untuk komputer *virtual plant*.



**Gambar 2.13** Respon sistem di komputer kontroler



**Gambar 2.14** Respon sistem di komputer *virtual plant*

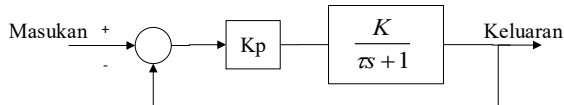
Sistem dengan kontroler PI *self-tuning* yang diterapkan pada dua komputer memiliki respon yang sama dengan pada saat diimplementasikan pada satu komputer. Akan tetapi, terjadi *delay* antara komputer kontroler dan komputer *virtual plant*. Hal tersebut terjadi karena adanya proses konversi sinyal dari digital menjadi analog dan sebaliknya pada saat proses pengiriman sinyal. Berdasarkan eksperimen yang dilakukan, pengaturan periode *sampling* yang sesuai dapat mengatasi *delay* yang terjadi pada sistem tersebut.

## LAMPIRAN

### 1. Perhitungan Perancangan Kontroler PI dengan Metode Analitik

Terdapat beberapa metode perancangan kontroler PI yang dapat digunakan. Pada penelitian ini dipilih metode analitik untuk diterapkan. Dengan metode tersebut, nilai parameter  $K_p$  dan  $\tau$ , diperoleh melalui proses berikut.

#### Kontroler Proporsional



$$\frac{Y(s)}{X(s)} = \frac{Kp \left( \frac{K}{\tau s + 1} \right)}{1 + Kp \left( \frac{K}{\tau s + 1} \right)}$$

$$\frac{Y(s)}{X(s)} = \frac{\frac{K.Kp}{\tau s + 1}}{\frac{\tau s + 1 + K.Kp}{\tau s + 1}}$$

$$\frac{Y(s)}{X(s)} = \frac{K.Kp}{\tau s + 1 + K.Kp}$$

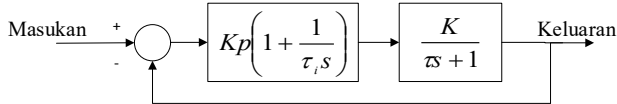
$$\frac{Y(s)}{X(s)} = \frac{\frac{K.Kp}{1 + K.Kp}}{\frac{\tau}{1 + K.Kp} s + 1}$$

$$\tau^* = \frac{\tau}{1 + K.Kp}$$

$$Kp = \frac{1}{K} \left( \frac{\tau}{\tau^*} - 1 \right)$$

### Kontroler Proporsional Integrator

Ada 2 kondisi pengaruh penambahan integrator pada respon sistem, yaitu apabila diberikan  $\tau_i = \tau$  maka respon sistem berupa orde pertama. Jika diberikan  $\tau_i \neq \tau$  maka respon sistem berupa respon orde kedua. Penjabarannya seperti berikut.



$$\frac{Y(s)}{X(s)} = \frac{Kp \left( 1 + \frac{1}{\tau_i s} \right) \left( \frac{K}{\tau s + 1} \right)}{1 + Kp \left( 1 + \frac{1}{\tau_i s} \right) \left( \frac{K}{\tau s + 1} \right)}$$

Jika dipilih  $\tau_i = \tau$ , maka :

$$\frac{Y(s)}{X(s)} = \frac{Kp \left( \frac{\tau_i s + 1}{\tau_i s} \right) \left( \frac{K}{\tau_i s + 1} \right)}{1 + Kp \left( \frac{\tau_i s + 1}{\tau_i s} \right) \left( \frac{K}{\tau_i s + 1} \right)}$$

$$\frac{Y(s)}{X(s)} = \frac{\frac{K.Kp}{\tau_i s}}{1 + \frac{K.Kp}{\tau_i s}}$$

$$\frac{Y(s)}{X(s)} = \frac{1}{\frac{\tau_i}{K.Kp} s + 1} \text{ persamaan orde pertama}$$

Jika dipilih  $\tau_i \neq \tau$ , maka :



$$\frac{Y(s)}{X(s)} = \frac{Kp \left( 1 + \frac{1}{\tau_i s} \right) \left( \frac{K}{\tau s + 1} \right)}{1 + Kp \left( 1 + \frac{1}{\tau_i s} \right) \left( \frac{K}{\tau s + 1} \right)}$$

$$\frac{Y(s)}{X(s)} = \frac{Kp \left( 1 + \frac{1}{\tau_i s} \right) \left( \frac{K}{\tau s + 1} \right)}{1 + Kp \left( 1 + \frac{1}{\tau_i s} \right) \left( \frac{K}{\tau s + 1} \right)} x \frac{\frac{1}{Kp \left( 1 + \frac{1}{\tau_i s} \right) \left( \frac{K}{\tau s + 1} \right)}}{\frac{1}{Kp \left( 1 + \frac{1}{\tau_i s} \right) \left( \frac{K}{\tau s + 1} \right)}}$$

$$\frac{Y(s)}{X(s)} = \frac{1}{\frac{1}{Kp \left( 1 + \frac{1}{\tau_i s} \right) \left( \frac{K}{\tau s + 1} \right)} + 1}$$

$$\frac{Y(s)}{X(s)} = \frac{\tau_i s + 1}{\frac{\tau_i \cdot \tau}{K \cdot Kp} s^2 + \left( \frac{\tau_i (1 + K \cdot Kp)}{K \cdot Kp} \right) s + 1} \text{ persamaan orde 2}$$

## 2. Perhitungan Metode Estimasi *Least Squares*

Model linier diskrit untuk sistem orde pertama :

$$y(k) - ay(k-1) = bu(k)$$

$$y(k) = ay(k-1) + bu(k)$$

Model hasil estimasi pasti memiliki kesalahan terhadap model plant asli, sehingga dirumuskan :

$$Y = \varphi\theta + E$$

$$y(k) = [y(k-1) \quad u(k)] \begin{bmatrix} a \\ b \end{bmatrix}$$

$$E = Y - \varphi\theta$$

Karena nilai kesalahan bisa positif maupun negatif, maka nilai tersebut dikuadratkan agar hasilnya positif.

$$E^T E = (Y - \varphi\theta)^T (Y - \varphi\theta)$$

$$E^T E = Y^T Y - \varphi^T Y \theta + Y^T \varphi \theta + \varphi^T \varphi \theta^2$$

$$E^T E = Y^T Y - 2\varphi^T Y \theta + \varphi^T \varphi \theta^2$$

Diharapkan diperoleh model dengan nilai kesalahan 0. Dipilih  $\theta$  untuk meminimalkan kesalahan tersebut.

$$\frac{\partial(E^T E)}{\partial \theta} = 0$$

$$\frac{\partial(E^T E)}{\partial \theta} = -2\varphi^T Y + 2\varphi^T \varphi \theta = 0$$

$$2\varphi^T Y = 2\varphi^T \varphi \theta$$

$$\theta = (\varphi^T \varphi)^{-1} \varphi^T Y$$

### 3. Formulasi Perancangan Kontroler

Formulasi kontroler PI ditentukan oleh model estimasinya. Dari model estimasi dengan metode *Least Squares* diperoleh  $\theta^T = [a_1 \dots a_{na} b_0 \dots b_{nb}]$ , sehingga dengan nilai a dan b tersebut dapat digunakan untuk merancang kontroler baru.

Model orde-1 diskrit ditunjukkan pada penurunan rumus berikut :

$$\frac{Y(s)}{X(s)} = \frac{K}{\tau s + 1} \quad \text{dimana} \quad s = \frac{2}{Ts} \left( \frac{1 - z^{-1}}{1 + z^{-1}} \right)$$

$$\frac{Y(z)}{X(z)} = \frac{K}{\frac{2\tau}{Ts} \left( \frac{1 - z^{-1}}{1 + z^{-1}} \right) + 1}$$

$$\frac{Y(z)}{X(z)} = \frac{K(1+z^{-1})}{\frac{2\tau}{T_s}(1-z^{-1}) + (1+z^{-1})}$$

$$\frac{Y(z)}{X(z)} = \frac{K(1+z^{-1})}{\left(1 + \frac{2\tau}{T_s}\right) + \left(1 - \frac{2\tau}{T_s}\right)z^{-1}}$$

$$\left(1 + \frac{2\tau}{T_s}\right)y(k) + \left(1 - \frac{2\tau}{T_s}\right)y(k-1) = kx(k) + kx(k-1)$$

$$y(k) = \frac{1}{1 + \frac{2\tau}{T_s}} \left\{ -\left(\frac{2\tau}{T_s} - 1\right)y(k-1) + kx(k-1) \right\}$$

$$y(k) = -\frac{2\tau - T_s}{2\tau + T_s} y(k-1) + \frac{kT_s}{2\tau + T_s} x(k-1)$$

Bentuk umum diskrit orde pertama :

$$y(k) = -a_1 y(k-1) + b_0 x(k-1)$$

$$a_1 = \frac{2\tau - T_s}{2\tau + T_s}$$

$$(2a_1 - 2)\tau = -(a_1 + 1)T_s$$

$$\tau = -\frac{a_1 + 1}{2(a_1 - 1)} T_s$$

$$K = (2\tau + T_s) \frac{b_0}{T_s}$$

## DAFTAR PUSTAKA

- [1]. Yu Jian, Liu Changliang. ***“Design of Self-tuning PID Controller with Fuzzy Variable Parameter Based on LabView”***. IEEE International Conference on Information and Automation. Pages 2586-2591. 2015
- [2]. M H Moradi. ***“New Techniques for PID Controller Design”***. IEEE International Conference on Control Automation. Vol : 2, pages 903 – 908.2003
- [3]. Karl J. Astrom, Bjorn Wittenmark. ***“Adaptive Control”***. Prentice Hall PTR. 1999.
- [4]. Takeshi Iwasa, Noboru Morizumi, Sigeru Omatu. ***“Temperature Control in a Batch Process by Neural Networks”***. IEEE International Conference on Vol:4, pages 2430 – 2433. 1997.
- [5]. Jiang Wei. ***“Reasearch on the Temperature Control System Based on Fuzzy Self-tuning PID”***. International Conference on Computer Design and Applications (ICCDA). Vol:3. 2010.
- [6]. Halvorsen Hans-Petter. ***“Hardware-in-the-Loop Simulation”***. Department of Electrical Engineering, Information Technology and Cybernetics. Telemark University College. 2012.
- [7]. Pavel Brandsetter, Marek Dobrovsky, Petr Krna. ***“Hardware in the Loop Simulation of A.C. Drive with the Induction Motor”***. IEEE International Conference on Applied Electronics. 2011.
- [8]. Tzouanas Vassilios, Stevenson Matthew. ***“Temperature and Level Control of a Multivariable Water Tank Process”***. 120<sup>th</sup> ASEE Annual Conference & Exposition, American Society for Engineering Education. 2013.
- [9]. Karl J. Åström and Tore Hägglund. ***“PID Controllers: Theory, Design, and Tuning 2<sup>nd</sup> edn”***. Instrument Society of America. Research Triangle Park, NC. 1995.
- [10]. Babatunde A. Ogunnaike and W. Harmon Ray. ***“Process Dynamics, Modeling and Control”***. Oxford University Press. 1994.
- [11]. Sigurd Skogestad. ***“Chemical and Energy Process Engineering”***. CRC Press Taylor and Francis Group. 2009.
- [12]. Lennart Ljung. ***“System Identification”***. Prentice Hall PTR. 1999.

- [13]. Dogan Ibrahim. ***“Microcontroller Based Applied Digital Control”***. Department of Computer Engineering Near East University, Cyprus. 2006.
- [14]. Hans-Petter Halvorsen. ***“Control and Simulation in LabVIEW”***. Department of Electrical Engineering, Information Technology and Cybernetics, Telemark University College. 2011.
- [15]. Goran S. Miljkovic, Ivana S. Stojkovic and Dragan B. Denic. ***“Generation and Application of Pseudo Random Binary Sequences Using Virtual Instrumentation”***. Department of Measurement, Faculty of Electronic Engineering, University of Nis. 2011.
- [16]. T. Chai and R. R. Draxler. ***“Root Mean Square Error (RMSE) or Mean Absolute Error (MAE) Argument Against Avoiding RMSE in the Literature”***. Technical Report. University Research Court, College Park, USA. 2014.
- [17]. Timothy L. Johnson, Charles A. Harvey and Gunter Stein. ***“Self-Tuning Regulator Design for Adaptive Control of Aircraft Wing/Store Flutter”***. IEEE Transaction on Automatic Control. 1982.

## RIWAYAT PENULIS



Puji Rahayu lahir di Banyuwangi, 16 Desember 1992. Merupakan anak kedua dari 3 bersaudara pasangan Bapak Murtiono dan Ibu Tasmi. Penulis menamatkan pendidikan Sekolah Dasar di SDN 2 Gendoh, SMPN 1 Sempu, dan SMKN 1 Glagah. Setelah menamatkan jenjang SMK, penulis melanjutkan kuliah D3 Teknik Elektronika di Politeknik Elektronika Negeri Surabaya (PENS). Setelah tamat dari D3 penulis melanjutkan pendidikan di S1 Teknik Elektro di Institut Teknologi Sepuluh Nopember (ITS) dengan mengambil konsentrasi Teknik Sistem Pengaturan. Pada tanggal 15 Juni 2016, penulis mengikuti ujian Tugas Akhir sebagai syarat untuk menyelesaikan masa studi dan memperoleh gelar Sarjana Teknik Elektro.

*E-mail* : [pujirahayu1612@gmail.com](mailto:pujirahayu1612@gmail.com)